A LETTERS JOURNAL EXPLORING THE FRONTIERS OF PHYSICS

March 2008

EPL, **81** (2008) 58004 www.epljournal.org
doi: 10.1209/0295-5075/81/58004

Effect of initial configuration on network-based recommendation

T. ZHOU[1,2,3(a)], L.-L. JIANG[2], R.-Q. SU[2,3] and Y.-C. ZHANG[1,3(b)]

[1] *Department of Physics, University of Fribourg - Chemin du Muse 3, CH-1700 Fribourg, Switzerland*
[2] *Department of Modern Physics and Nonlinear Science Center, University of Science and Technology of China Hefei Anhui, 230026, PRC*
[3] *Information Economy and Internet Research Laboratory, University of Electronic Science and Technology of China - Chengdu Sichuan, 610054, PRC*

received 15 November 2007; accepted in final form 15 January 2008
published online 13 February 2008

PACS 89.75.Hc – Networks and genealogical trees
PACS 87.23.Ge – Dynamics of social systems
PACS 05.70.Ln – Nonequilibrium and irreversible thermodynamics

Abstract – In this paper, based on a weighted object network, we propose a recommendation algorithm, which is sensitive to the configuration of initial resource distribution. Even under the simplest case with binary resource, the current algorithm has remarkably higher accuracy than the widely applied global ranking method and collaborative filtering. Furthermore, we introduce a free parameter β to regulate the initial configuration of resource. The numerical results indicate that decreasing the initial resource located on popular objects can further improve the algorithmic accuracy. More significantly, we argue that a better algorithm should simultaneously have higher accuracy and be more personal. According to a newly proposed measure about the degree of personalization, we demonstrate that a degree-dependent initial configuration can outperform the uniform case for both accuracy and personalization strength.

Introduction. – The exponential growth of the Internet [1] and World-Wide-Web [2] confronts people with an information overload: they are facing too many data and sources to be able to find out those most relevant for them. Thus far, the most promising way to efficiently filter out the information overload is to provide personal recommendations. That is to say, using the personal information of a user (*i.e.*, the historical track of this user's activities) to uncover his habits and to consider them in the recommendation. For instances, `Amazon.com` uses one's purchase history to provide individual suggestions. If you have bought a textbook on statistical physics, Amazon may recommend you some other statistical-physics books. Based on the well-developed *Web 2.0* technology, recommendation systems are frequently used in web-based movie-sharing (music-sharing, book-sharing, etc.) systems, web-based selling systems, and so on. Motivated by the significance to the economy and society, recommendation algorithms are being extensively investigated in the engineering community [3]. Various kinds of algorithms have been proposed, including correlation-based methods [4,5], content-based methods [6,7], the spectral analysis [8], principle component analysis [9], and so on.

Very recently some physical dynamics, including heat conduction process [10] and mass diffusion [11,12], have found applications in personal recommendation. These physical approaches have been demonstrated to be both highly efficient and of low computational complexity [10–12]. In this paper, we introduce a network-based recommendation algorithm with degree-dependent initial configuration. Compared with the uniform initial configuration, the prediction accuracy can be remarkably enhanced by using the degree-dependent configuration. More significantly, besides the prediction accuracy, we present novel measurements to judge how personal the recommendation results are. The algorithm providing more personal recommendations has, in principle, greater ability to uncover the individual habits. Since mainstream interests are more easily uncovered, a user may appreciate a system more if it can recommend the unpopular objects he/she enjoys. Therefore, we argue that those two kinds of measurements, accuracy and degree of personalization, are complementary to each other in evaluating a recommendation algorithm. Numerical simulations show that the optimal initial configuration subject to accuracy can also generate more personal recommendations.

(a) E-mail: zhutou@ustc.edu
(b) E-mail: yi-cheng.zhang@unifr.ch

Method. – A recommendation system consists of users and objects, and each user has collected some objects. Denoting the object set as $O=\{o_1,o_2,\cdots,o_n\}$ and the user set as $U=\{u_1,u_2,\cdots,u_m\}$, the recommendation system can be fully described by an $n\times m$ adjacent matrix $A=\{a_{ij}\}$, where $a_{ij}=1$ if o_i is collected by u_j, and $a_{ij}=0$ otherwise. A reasonable assumption is that the objects you have collected are what you like, and a recommendation algorithm aims at predicting your personal opinions (to what extent you like or hate them) on those objects you have not yet collected. Mathematically speaking, for a given user, a recommendation algorithm generates a ranking of all the objects he/she has not collected before. The top L objects are recommended to this user, with L the length of the recommendation list.

Based on the user-object relations A, an object network can be constructed, where each node represents an object, and two objects are connected if and only if they have been collected simultaneously by at least one user. We assume a certain amount of resource (*e.g.*, recommendation power) is associated with each object, and the weight w_{ij} represents the proportion of the resource o_j would like to distribute to o_i. For example, in the book-selling system, the weight w_{ij} contributes to the strength of book o_i recommendation to a customer provided he has bought book o_j. Following a network-based resource-allocation process where each object distributes its initial resource equally to all the users who have collected it, and then each user sends back what he/she has received to all the objects he/she has collected (also equally), the weight w_{ij} (the fraction of initial resource o_j eventually gives to o_i) can be expressed as:

$$w_{ij}=\frac{1}{k(o_j)}\sum_{l=1}^{m}\frac{a_{il}a_{jl}}{k(u_l)}, \tag{1}$$

where $k(o_j)=\sum_{i=1}^{n}a_{ji}$ and $k(u_l)=\sum_{i=1}^{m}a_{il}$ denote the *degrees* of object o_j and u_l, respectively. Clearly, the weight between two unconnected objects is zero. According to the definition of the weighted matrix $W=\{w_{ij}\}$, if the initial resource vector is $\mathbf{f}$, the final resource distribution is $\mathbf{f}'=W\mathbf{f}$.

The general framework of the proposed network-based recommendation is as follows: i) construct the weighted object network (*i.e.* determine the matrix W) from the known user-object relations; ii) determine the initial resource vector $\mathbf{f}$ for each user; iii) get the final resource distribution via $\mathbf{f}'=W\mathbf{f}$; iv) recommend those uncollected objects with highest final resource. Note that the initial configuration $\mathbf{f}$ is determined by the user's personal information, thus for different users, the initial configuration is different. From now on, for a given user u_i, we use $\mathbf{f}^i$ to emphasize this personal configuration.

Actually, the present method can also be extended to a diffusion-like algorithm, which focus on the stationary solution of equation $\mathbf{f}^*=W\mathbf{f}^*$ where a few boundary elements can be different (see the details in a similar method for multi-rating system [12]). However, as we pointed out (see fig. 2 in ref. [12]), the diffusion method greatly increases the computational complexity, but cannot provide more accurate recommendation results compared with the present one. Therefore, in this paper we will not discuss the extension to a diffusion-like algorithm.

Numerical results. – For a given user u_i, the j-th element of $\mathbf{f}^i$ should be zero if $a_{ji}=0$. That is to say, one should not put any recommendation power (*i.e.* resource) onto an uncollected object. The simplest case is to set a uniform initial configuration as

$$f_j^i=a_{ji}. \tag{2}$$

Under this configuration, all the objects collected by u_i have the same recommendation power. In spite of its simplicity, it can outperform the two most widely applied recommendation algorithms, the *global ranking method* (GRM)[1] and the *collaborative filtering* (CF)[2]. A more detailed discussion on collaborative filtering can be found in refs. [4,5,13,14].

To test the algorithmic accuracy, we use a benchmark data set, namely *MovieLens*[3]. The data consists of 1682 movies (objects) and 943 users, and users vote movies using discrete ratings 1–5. We therefore applied a coarse-graining method similar to that used in ref. [15]: a movie has been collected by a user if and only if the giving rating is at least 3 (*i.e.* the user at least likes this movie). The original data contains 10^5 ratings, 85.25% of which are $\geqslant 3$, thus after coarse gaining the data contains 85250 user-object pairs. To test the recommendation algorithms, the data set is randomly divided into two parts: The training set contains 90% of the data, and the remaining 10% of data constitutes the probe. The training set is treated as known information, while no information in the probe set is allowed to be used for prediction.

A recommendation algorithm should provide each user with an ordered queue of all its uncollected objects. For an arbitrary user u_i, if the relation u_i-o_j is in the probe set (according to the training set, o_j is an uncollected object for u_i), we measure the position of o_j in the ordered queue. For example, if there are 1000 uncollected movies for u_i, and o_j is the 10th from the top, we say the position of o_j is 10/1000, denoted by $r_{ij}=0.01$. Since the probe entries are actually collected by users, a good algorithm is expected to give high recommendations to them, thus leading to small r. Therefore, the mean

[1]The global ranking method sorts all the objects in the descending order of degree and recommends those with highest degrees.

[2]The collaborative filtering is based on measuring the similarity between users. For two users u_i and u_j, their similarity can be simply determined by $s_{ij}=\sum_{l=1}^{n}a_{li}a_{lj}/\min\{k(u_i),k(u_j)\}$. For any user-object pair u_i-o_j, if u_i has not yet collected o_j (*i.e.*, $a_{ji}=0$), the predicted score, v_{ij} (to what extent u_i likes o_j), is given as $v_{ij}=\sum_{l=1,l\neq i}^{m}s_{li}a_{jl}/\sum_{l=1,l\neq i}^{m}s_{li}$. For any user u_i, all the nonzero v_{ij} with $a_{ji}=0$ are sorted in descending order, and those objects in the top are recommended.

[3]The MovieLens data can be downloaded from the web-site of *GroupLens Research* (`http://www.grouplens.org`).

value of the position value $\langle r \rangle$ (called *ranking score* [11]), averaged over all the entries in the probe, can be used to evaluate the algorithmic accuracy: the smaller the ranking score, the higher the algorithmic accuracy, and vice versa. Implementing the three algorithms mentioned above, the average values of ranking scores over five independent runs (one run here means an independently random division of data set) are 0.107, 0.122, and 0.140 for network-based recommendation, collaborative filtering, and global ranking method, respectively. Clearly, even under the simplest initial configuration, subject to the algorithmic accuracy, the network-based recommendation outperforms the other two algorithms.

Consider the initial resource located on the object o_i as its assigned recommendation power. In the whole recommendation process, the total power given to o_i is $p_i = \sum_j f_i^j$, where the superscript j runs over all the users u_j. Under uniform initial configuration (see eq. (2)), the total power of o_i is $p_i = \sum_j f_i^j = \sum_j a_{ij} = k(o_i)$. That is to say, the total recommendation power assigned to an object is proportional to its degree, thus the impact of high-degree objects (*e.g.*, popular movies) is enhanced. Although it already has a good algorithmic accuracy, this uniform configuration may be oversimplified, and depressing the impact of high-degree objects in an appropriate way could, perhaps, further improve the accuracy. Motivated by this, we propose a more complicated distribution of initial resource to replace eq. (2):

$$f_j^i = a_{ji} k^{\beta}(o_j), \tag{3}$$

where β is a tunable parameter. Compared with the uniform case, $\beta = 0$, a positive β strengthens the influence of large-degree objects, while a negative β weakens the influence of large-degree objects. In particular, the case $\beta = -1$ corresponds to an identical allocation of recommendation power ($p_i = 1$) for each object o_i.

Figure 1 reports the algorithmic accuracy as a function of β. The curve has a clear minimum around $\beta = -0.8$. Compared with the uniform case, the ranking score can be further reduced by 9% at the optimal value. It is indeed a great improvement for recommendation algorithms. Note that $\beta_{\rm opt}$ is close to -1, which indicates that the more homogeneous distribution of recommendation power among objects may lead to a more accurate prediction.

Besides accuracy, another significant ingredient one should take into account for a personal recommendation algorithm is how personal this algorithm is. For example, suppose there are 10 perfect movies not yet known for user u_i, 8 of which are widely popular, while the other two fit a certain specific taste of u_i. An algorithm recommending the 8 popular movies is very nice for u_i, but he may feel even better about a recommendation list containing those two unpopular movies. Since there are countless channels to obtain information on popular movies (TV, the Internet, newspapers, radio, etc.), uncovering very specific preference, corresponding to unpopular objects, is much more significant than simply picking out what a user likes from the top of the list. To measure this factor, we go simultaneously in two directions. Firstly, given the length L of recommendation list, the popularity can be measured directly by averaging the degree $\langle k \rangle$ over all the recommended objects. One can see from fig. 2 that the average degree is positively correlated with β, thus depressing the recommendation power of high-degree objects gives more opportunity to unpopular objects. Also for $L = 10$, 50 and 100, the corresponding $\langle k \rangle$ are 353.50, 258.00 and 214.09 (GRM), as well as 84.62, 87.95 and 83.79 (CF). Since GRM always recommends the most popular objects, it is clear that $\langle k \rangle_{\rm GRM}$ is the largest. On the other hand, CF mainly depends the similarity between users. Thus one user may be recommended an object collected by another

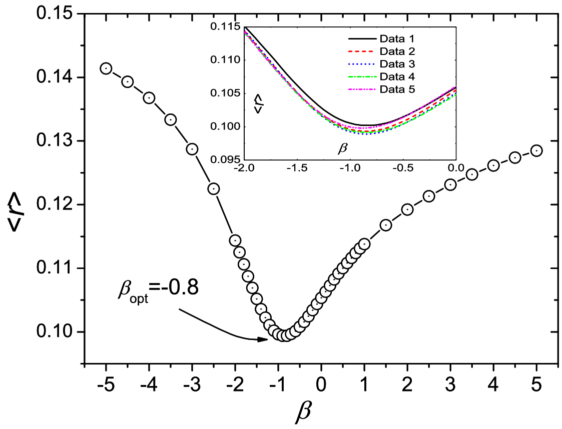

Fig. 1: (Color online) The ranking score $\langle r \rangle$ *vs.* β. The optimal β, corresponding to the minimal $\langle r \rangle \approx 0.098$, is $\beta_{\rm opt} \approx -0.8$. All the data points shown in the main plot is obtained by averaging over five independent runs with different data set divisions. The inset shows the numerical results of every separate run, where each curve represents one random division of data set.

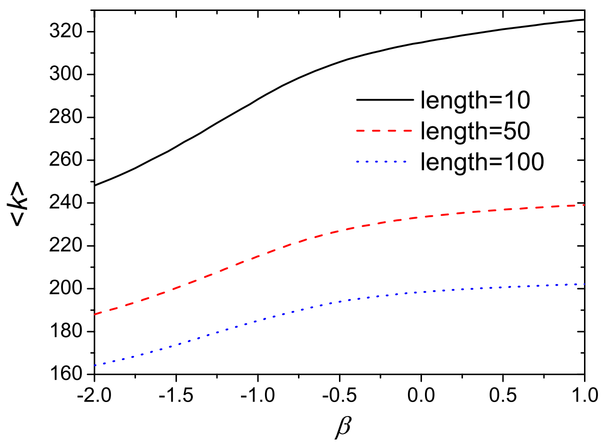

Fig. 2: (Color online) The average degree of all recommended movies *vs.* β. The black solid, red dashed and blue dotted curves represent the cases with typical lengths $L = 10$, 50 and 100, respectively. All the data points are obtained by averaging over five independent runs with different data set divisions.

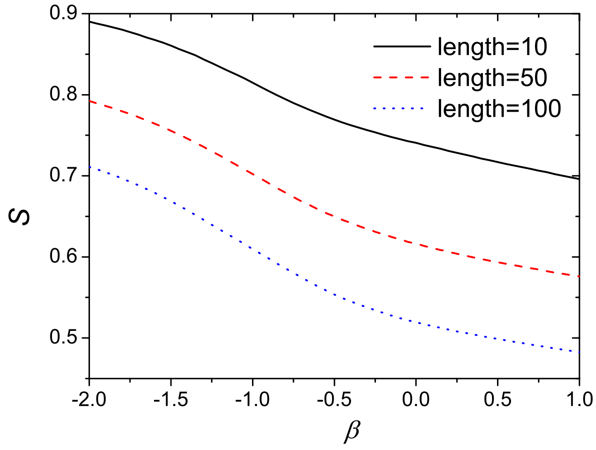

Fig. 3: (Color online) S *vs.* β. The black solid, red dashed and blue dotted curves represent the cases with typical lengths $L = 10$, 50 and 100, respectively. All the data points are obtained by averaging over five independent runs with different data set divisions.

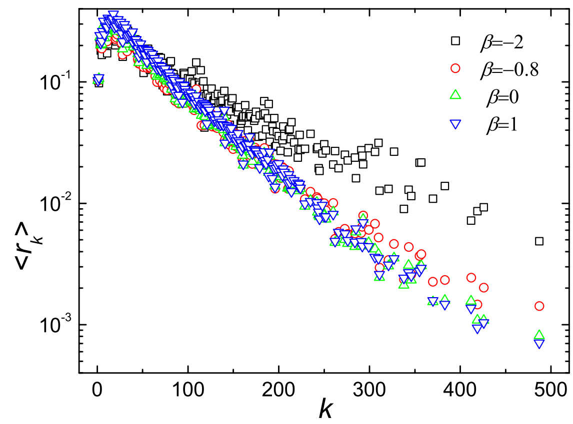

Fig. 4: (Color online) A scatter plot showing the average ranking score, $\langle r_k \rangle$, and the object degree, k. The black squares, red circles, green up-triangles and blue down-triangles represent the cases of $\beta = -2$, $\beta = -0.8$, $\beta = 0$ and $\beta = 1$, respectively. $\beta = -0.8$ corresponds to the optimal case. For a clear observation, the y-axis is set to be logarithmic. All the data points are obtained by averaging over five independent runs with different data set divisions.

user having very similar habits to him, even though this object may be very unpopular. This is the reason why $\langle k \rangle_{\mathrm{CF}}$ is the smallest. Secondly, one can measure the strength of personalization via the Hamming distance. If the overlapped number of objects in u_i and u_j's recommendation lists is Q, their Hamming distance is $H_{ij} = 1 - Q/L$. Generally speaking, a more personal recommendation list should have larger Hamming distances to other lists. Accordingly, we use the mean value of Hamming distance $S = \langle H_{ij} \rangle$, averaged over all the user-user pairs, to measure the strength of personalization. Figure 3 plots S *vs.* β and, in accordance with the numerical results shown in fig. 2, depressing the influence of high-degree objects makes the recommendations more personal. For $L = 10$, 50 and 100, the corresponding S are 0.508, 0.397 and 0.337 (GRM), as well as 0.654, 0.501 and 0.421 (CF). Note that, S_{GRM} is obviously larger than zero, because the collected objects will not appear in the recommendation list, thus different users have different recommendation lists. Since CF has the potential to enhance the user-user similarity, S_{CF} is remarkably smaller than that corresponding to negative β in network-based recommendation.

In a word, without any increase in the algorithmic complexity, using an appropriate negative β in our algorithm outperforms the uniform case (*i.e.* $\beta = 0$) for all three criteria: more accurate, less popular, and more personalized.

Dependence of algorithmic accuracy on the object degree. – The ranking score reported in fig. 1 provides us with a macroscopic description of the algorithmic accuracy. To make clear the role of β, a microscopic understanding of algorithmic accuracy is very helpful. Especially, since β is a regulating parameter on the object degree, we would like to see the dependence of accuracy on the object degree. Given an object degree k, the average ranking score, denoted by $\langle r_k \rangle$, is defined as the mean value of the position averaged over all the entries in the probe with object degree equal to k.

Figure 4 reports the correlation between accuracy and object degree. Clearly, the algorithm is more accurate for popular (large degree) objects. The decay of the ranking score approximately obeys an exponential form. Actually, the algorithm with smaller β has a better accuracy for unpopular objects, while the one with larger β performs better for popular objects. Therefore, the best accuracy is corresponding to a proper β, namely the optimal β. For the current data set, it is about -0.8. Besides the overall tendency (negative correlation between $\langle r_k \rangle$ and k), fig. 4 displays a small peak at $k_p \sim 20$. A possible reason is, the selections of extremely unpopular movies probably stand for some specific tastes different from the mainstream, which are easier to be found out.

To see the performance of the current algorithm in some extreme circumstances, we sort all the entries in the probe by a descending order of object degree. That is to say, the user-object pair put in the top is the one corresponding to the most popular objects. Figures 5(a) and (b) report the average ranking score for the top-1000 entries and the bottom-1000 entries, respectively. As shown in fig. 5(a), the ranking score shows a monotonously negative relation with β, meaning the larger β outperforms smaller β for the popular movies, which is in accordance with the scatter plot in fig. 4. For the unpopular movies, as shown in fig. 5(b), the curve displays a minimum around $\beta = -2$, which is remarkably smaller than the optimal value, $\beta = -0.8$, for the overall data set. Actually, the more unpopular the probe set we choose, the smaller the optimal value of β.

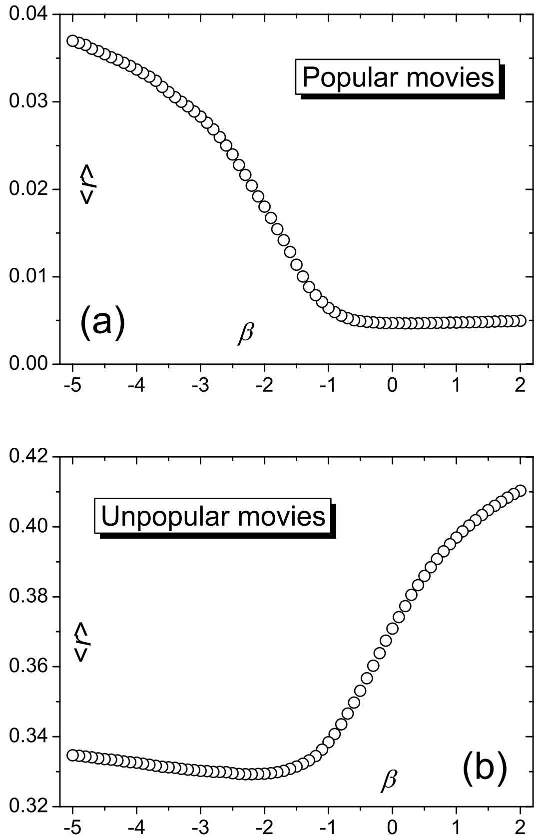

Fig. 5: The ranking score $\langle r \rangle$ *vs.* β in two extreme circumstances: (a) taking into account only the average ranking score of 1000 entries in the probe with the highest object degrees; (b) taking into account only the average ranking score of 1000 entries in the probe with the lowest object degrees.

Conclusions. – In this paper, we propose a recommendation algorithm based on a weighted object network. This algorithm is sensitive to the configuration of the initial resource distribution. Even under the simplest case with binary resource, the current algorithm has remarkably higher accuracy than the widely applied GRM and CF. Since the computational complexity of this algorithm is much less than that of CF[4], it has great potential significance in practice. Furthermore, we introduce a free parameter β to regulate the initial configuration of resource. Numerical results indicate that decreasing the initial resource located on popular objects further improves the algorithmic accuracy: In the optimal case ($\beta_{\rm opt} \approx -0.8$), the distribution of total initial resource located on each object is very homogeneous ($p_i \sim k^{0.2}(o_i)$). Besides the ranking score, there have been many measures suggested to evaluate the accuracy of personal recommendation algorithms [11,16–18], including *hitting rate*, *precision*, *recall*, *F-measure*, and so on. However, thus far, there has been no consideration of the degree of personalization. In this paper, we suggest two measures, $\langle k \rangle$ and S, to address this issue. We argue that to evaluate the performance of a recommendation algorithm, one should take into account not only the accuracy, but also the degree of personalization and popularity of recommended objects. Even under this more strict criterion, the case with $\beta_{\rm opt} \approx -0.8$ outperforms the uniform case. Theoretical physics provides us some beautiful and powerful tools in dealing with this long-standing challenge in modern information science: how to do a personal recommendation. We believe the current work can enlighten readers in this interesting direction.

[4]Instead of calculating all the elements in W, one can implement the current algorithm by directly diffusing the resource of each user. Ignoring the degree-degree correlation in user-object relations, the algorithmic complexity is $\mathbb{O}(m\langle k_u \rangle \langle k_o \rangle)$, where $\langle k_u \rangle$ and $\langle k_o \rangle$ denote the average degree of users and objects. Correspondingly, the algorithmic complexity of collaborative filtering is $\mathbb{O}(m^2\langle k_u \rangle + mn\langle k_o \rangle)$, where the first term accounts for the calculation of similarity between users, and the second term accounts for the calculation of the predictions. For GRM, since it only requires a sorted list of objects by their degrees, its algorithmic complexity is simply $\mathbb{O}(n \log n)$.

***

We acknowledge Runran Liu for very valuable discussion and comments on this work. This work is partially supported by SBF (Switzerland) for financial support through project C05.0148 (Physics of Risk), and the Swiss National Science Foundation (205120-113842). TZ acknowledges NNSFC under Grant No. 10635040 and 60744003, as well as the 973 Program 2006CB705500.

REFERENCES

[1] Faloutsos M., Faloutsos P. and Faloutsos C., *Comput. Commun. Rev.*, **29** (1999) 251.
[2] Broder A., Kumar R., Moghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A. and Wiener J., *Comput. Netw.*, **33** (2000) 309.
[3] Adomavicius G. and Tuzhilin A., *IEEE Trans. Knowl. Data Eng.*, **17** (2005) 734.
[4] Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R. and Riedl J., *Commun. ACM*, **40** (1997) 77.
[5] Herlocker J. L., Konstan J. A., Terveen K. and Riedl J. T., *ACM Trans. Inf. Syst.*, **22** (2004) 5.
[6] Balabanović M. and Shoham Y., *Commun. ACM*, **40** (1997) 66.
[7] Pazzani M. J., *Artif. Intell. Rev.*, **13** (1999) 393.
[8] Maslov S. and Zhang Y.-C., *Phys. Rev. Lett.*, **87** (2001) 248701.
[9] Goldberg K., Roeder T., Gupta D. and Perkins C., *Inf. Retr.*, **4** (2001) 133.
[10] Zhang Y.-C., Blattner M. and Yu Y.-K., *Phys. Rev. Lett.*, **99** (2007) 154301.
[11] Zhou T., Ren J., Medo M. and Zhang Y.-C., *Phys. Rev. E*, **76** (2007) 046115.
[12] Zhang Y.-C., Medo M., Ren J., Zhou T., Li T. and Yang F., *EPL*, **80** (2007) 68003.
[13] Sarwar B., Karypis G., Konstan J. and Riedl J., *Proceedings of the 10th International Conference WWW*, published on CD, 2001, pp. 285–295.
[14] Liu R.-R., Jia C.-X., Zhou T., Sun D. and Wang B.-H., arXiv:0801.1333.

[15] Blattner M., Zhang Y.-C. and Maslov S., *Physica A*, **373** (753) 2007.
[16] Billsus D. and Pazzani M. J., *Proceedings of the 15th International Conference Machine Learning* (Morgan Kaufmann Publishers, San Francisco) 1998, pp. 46–54.
[17] Sarwar B., Karypis G., Konstan J. and Riedl J., *Proceedings of the ACM Conference Electronic Commerce* (ACM Press) 2000, pp. 158–167.
[18] Huang Z., Chen H. and Zeng D., *ACM Trans. Inf. Syst.*, **22** (116) 2004.